FNLP Tutorial 2

Question 1. Evaluating data annotations

Imagine that this is your small corpus of named entities in a simple task, where we ignore a named entity's type and annotate the real named entities with square brackets:

> [Paris Hilton] stayed at the [Hilton] in [Paris] and
> [James Clerk Maxwell] was educated at [Edinburgh] and [Cambridge]

We formalise the annotation of a single sentence as a set A. Each element $a \in A$ represents a named entity as a span through an ordered pair of zero-based indices (a named entity $\langle a, b\rangle$ starts at position a and ends at b, including b). Assume a computational model predicts the following named entities: $\{\langle 0, 1\rangle, \langle 5, 5\rangle, \langle 9, 10\rangle, \langle 15, 15\rangle\}$

1. Compute the precision, recall, and F_1-score of this annotation.
2. Provide an annotation that would give a precision of more than 0.8 and a recall of less than 0.2, and use your answer to explain why the F_1-score uses the *harmonic* mean.
3. While the F_1-score is a better metric than precision and recall in isolation, there are other flaws all three metrics suffer from. What, specifically in the context of span identification, does it fail to capture about the model's predictions provided above?

Solution

1. There are 3 true positives, 1 false positive and 3 false negatives. Therefore, the precision is 0.75, the recall is 0.5, and the F_1-score is $2 \cdot \frac{0.75 \cdot 0.5}{0.75+0.5} = 0.6$.
2. We could, for example, use a simple annotation with perfect precision, such as $\{\langle 0, 1\rangle\}$ that gives a recall of $\frac{1}{6}$, and an F_1-score of approximately 0.286. This F_1-score lies far below the arithmetic mean of the two, and lies, relatively speaking, closer to the recall compared to (1) above. This illustrates why the harmonic mean is used: it penalises extreme values for either precision or recall. This is desirable since a trivial prediction – such as predicting all candidates, or no candidates at all – can yield a high recall or a high precision, but not both.
3. The span $\langle 9, 10\rangle$ is incorrect, but "James Clerk" is a part of a named entity. Nonetheless, this partial match between the prediction and the target is not captured by the current metrics.

Question 2. Text generation with a language model

We are given the following corpus, modified from J&M (Chapter 3 in the 3rd edition):

> <s> language is awesome </s>
> <s> language is awesome </s>
> <s> language and students </s>
> <s> awesome students like language </s>

1. Using a bigram language model without smoothing, generate 4 sentences, starting from '<s>'. To generate sentences manually, randomly choose a real number between 0 and 1, and use the cumulative probabilities of words in the vocabulary to select a word from the language model. For example, if there are two words in the vocabulary with non-zero conditional probabilities $P(\text{one}|\text{<s>}) = 0.25$ and $P(\text{two}|\text{<s>}) = 0.75$, then choosing a random number between 0 and 0.25 would result in 'one', whereas a random number between 0.25 and 1 would result in 'two'.

2. How does the lack of smoothing impact the novelty of the sentences generated?

3. How does the nature of the corpus impact the lengths of the generated sentences?

Prefix	**(Non-zero) Conditional probabilities**		
<s>	$P(\text{language}\|\text{<s>}) = \frac{3}{4}$	$P(\text{awesome}\|\text{<s>}) = \frac{1}{4}$	
and	$P(\text{students}\|\text{and}) = 1$		
awesome	$P(\text{</s>}\|\text{awesome}) = \frac{2}{3}$	$P(\text{students}\|\text{awesome}) = \frac{1}{3}$	
is	$P(\text{awesome}\|\text{is}) = 1$		
language	$P(\text{is}\|\text{language}) = \frac{1}{2}$	$P(\text{and}\|\text{language}) = \frac{1}{4}$	$P(\text{</s>}\|\text{language}) = \frac{1}{4}$
like	$P(\text{language}\|\text{like}) = 1$		
students	$P(\text{</s>}\|\text{students}) = \frac{1}{2}$	$P(\text{like}\|\text{students}) = \frac{1}{2}$	

Table 1: The non-zero conditional probabilities of the language model discussed in Exercise 2.1.

Solution

1. Table 1 provides the non-zero conditional probabilities that can be used to generate sentences for this question. Some example sentences that can be generated, are:

   - <s> awesome students like language and students like language </s>
   - <s> awesome students </s>
   - <s> language </s>
   - <s> language is awesome students like language </s>
   - <s> language is awesome </s>
   - <s> language </s>
   - <s> awesome </s>
   - <s> language </s>
   - <s> awesome </s>
   - <s> language and students like language is awesome students like language and students like language and students like language is awesome students like language </s>

2. The language model can generate sentences that are novel (i.e. that were not in the training corpus). In fact, 9 out of 10 example sentences above were not in the training corpus. However, every bigram does exist in the training corpus, so the language model is severely limited in the sentences it can generate.

3. The average lengths of the training corpus (5.25) and the generated sentences above (5.7) do not vary greatly, but the standard deviations—0.4 and 6.5, respectively—do. This is due to the presence of words that can appear in multiple positions in the sentence: 'awesome', 'language' and 'students'. As a result, a sentence can end after one word, or can restart when one would expect it to end.

Question 3. Smoothing

1. Compute $P(\text{awesome}|\text{is})$ and $P(\text{like}|\text{is})$ for (a) the bigram language model without smoothing, (b) with add-1 smoothing, and (c) using interpolation.
2. Which values of λ_1 and λ_2 did you select, and why?
3. As discussed in J&M section 3.5, one would normally select the values for λ_1 and λ_2 based on a held-out corpus. What are the characteristic features of a corpus that would result in $\lambda_2 \ll \lambda_1$?
4. Mention a disadvantage of using interpolation to do smoothing, by referring to the bigrams 'is awesome' and 'is like'. Identify another type of smoothing that overcomes the shortcomings you described.

Solution

1. The probabilities are provided below:

$$P_{\text{MLE}}(\text{awesome}|\text{is}) = \frac{C(\text{is, awesome})}{C(\text{is})} = \frac{2}{2} = 1 \tag{1}$$

$$P_{+1}(\text{awesome}|\text{is}) = \frac{C(\text{is, awesome}) + 1}{C(\text{is}) + V} = \frac{2+1}{2+8} = 0.3 \tag{2}$$

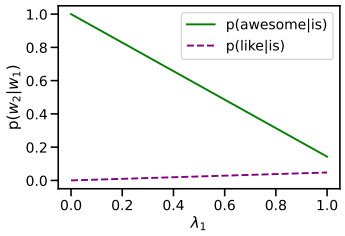
$$P_{\text{INT}}(\text{awesome}|\text{is}) = \lambda_1 P_{\text{MLE}}(\text{awesome}) + \lambda_2 P_{\text{MLE}}(\text{awesome}|\text{is}) = \lambda_1 \frac{3}{21} + \lambda_2 \frac{2}{2} \tag{3}$$

$$P_{\text{MLE}}(\text{like}|\text{is}) = \frac{C(\text{is, like})}{C(\text{is})} = \frac{0}{2} = 0 \tag{4}$$

$$P_{+1}(\text{like}|\text{is}) = \frac{C(\text{is, like}) + 1}{C(\text{is}) + V} = \frac{0+1}{2+8} = 0.1 \tag{5}$$

$$P_{\text{INT}}(\text{like}|\text{is}) = \lambda_1 P_{\text{MLE}}(\text{like}) + \lambda_2 P_{\text{MLE}}(\text{like}|\text{is}) = \lambda_1 \frac{1}{21} + \lambda_2 \frac{0}{2} \tag{6}$$

2. The graph illustrates the impact of different values for λ_1 and λ_2. There is not one correct setting, but to avoid stealing too much from seen events, one could set $\lambda_1 < 0.5$. On the other hand, if one's goal is to create a language model that is likely to generate events unseen during training, $\lambda_2 \ll \lambda_1$. Notice that with add-1 smoothing the language model is more likely to generate 'is like' than with interpolation smoothing, due to the low unigram probability of 'like'.



3. If the held-out corpus (on which we estimate our parameters) mostly contains bigrams from the training corpus, that would lead to $\lambda_2 \gg \lambda_1$: smoothing is hardly needed. On the other hand, a held-out corpus for which a lot of bigrams do not appear in the training corpus would lead to $\lambda_2 \ll \lambda_1$. Notice that including words in the held-out corpus that do not exist in the training corpus does not favour a higher or lower value for λ_1 or λ_2.
4. Linear interpolation linearly combines the unigram and bigram probabilities independent of the bigram that we compute probabilities for, while 'is awesome' appears in the training corpus and 'is like' does not. For the former bigram it might be more reliable to trust the bigram

language model, while for the latter one, we would like to borrow from the unigram model. A smoothing method that improves upon this is Katz backoff smoothing that relies on a discounted probability $P^*(w_2|w_1)$ if a bigram has been seen before, and relies on the unigram model, otherwise.

Question 4. Document classification

The following are features extracted from a set of short movie reviews, and the genre of the movie.

n	**Document**	**Class**
1	love, fast, romantic, couple	`romance`
2	romantic, fast, fun, fun	`comedy`
3	fast, violence, shoot, furious, fast	`action`
4	couple, fun, fast, fast, furious	`action`
5	fun, violence, fun, romantic	`comedy`
6	fast, fun	?

1. What is the MLE estimate of the prior probability $P(\texttt{action})$?
2. What is $P_{MLE}(\text{fast}|\texttt{action})$?
3. Apply add-α smoothing. Assuming that $\alpha = 0.7$, what is $P_{add-\alpha}(\text{fast}|\texttt{action})$?
4. Assuming d is the sixth document from the table above, compute the ratio $\frac{P(\texttt{comedy}|d)}{P(\texttt{action}|d)}$ with the same smoothing method applied. What does this ratio tell us about the document's classification?
5. Describe the type of classification model one would get for an extremely large value of α. Could changing α change the classification of our document number 6?

Solution

1. $\frac{2}{5} = 0.4$
2. $\frac{4}{10} = 0.4$
3. $\frac{4+0.7}{10+8\cdot 0.7} \approx 0.30$
4. $\frac{P(\texttt{comedy}|d)}{P(\texttt{action}|d)} = \frac{\frac{P(d|\texttt{c})P(\texttt{c})}{P(d)}}{\frac{P(d|\texttt{a})P(\texttt{a})}{P(d)}} = \frac{P(d|\texttt{c})P(\texttt{c})}{P(d|\texttt{a})P(\texttt{a})} = \frac{P(\text{fast}|\texttt{c})P(\text{fun}|\texttt{c})P(\texttt{c})}{P(\text{fast}|\texttt{a})P(\text{fun}|\texttt{a})P(\texttt{a})} = \frac{\frac{1.7}{13.6}\cdot\frac{4.7}{13.6}\cdot\frac{2}{5}}{\frac{4.7}{15.6}\cdot\frac{1.7}{15.6}\cdot\frac{2}{5}} = 1.32$
   This ratio indicates that the movie review is a bit more likely to be from a comedy than from an action movie.
5. As α grows, the likelihood of any feature for any class would approach $\frac{\alpha}{\alpha V}$, where V is our vocabulary size. The more equal the likelihood of any two features is, the larger the impact of the prior probability of the class on the classification. Eventually, the classifier would always predict the most frequent class.
   In this specific case, changing α would, however, not change the classification of our document due to equal prior probabilities of the classes `comedy` and `action`.

Bonus question

We said in the lecture that relative frequency estimation is a form of maximum likelihood estimation (MLE). Here we want you to *prove* that this is true for a categorical random variable.

Let X be a categorical random variable that can take values $1, \ldots, k$. Assume we have N independent samples $x_1, \ldots, x_N$ from X where each x_i takes one of the k values. We consider a family of distributions $\hat{P}_\theta(X = x) = \theta_x$ with parameters θ. The likelihood of the data as function of θ is:

$$L(\theta) = \prod_{i=1}^{N} \hat{P}_\theta(X = x_i) = \prod_{i=1}^{N} \theta_{x_i} \tag{7}$$

Show that $P_{MLE}(X = x) = \frac{C(x)}{N}$ where $C(x)$ is the count of x in our sample $x_1, \ldots, x_N$.

Hint you might want to use the following fact, known as Gibbs' inequality (see also Lecture 6b): If P and Q are distributions over the random variable X, then

$$-\sum_{i=1}^{k} P(X = i) \log P(X = i) \leq -\sum_{i=1}^{k} P(X = i) \log Q(X = i) \tag{8}$$

This holds with equality if and only if $P(X = i) = Q(X = i)$ for all i.

Solution Let us first rephrase the likelihood in terms of counts:

$$L(\theta) = \prod_{i=1}^{n} \hat{P}_\theta(X = x_i) = \prod_{x=1}^{k} \hat{P}_\theta(X = x)^{C(x)} \tag{9}$$

Note that the product now ranges over the values that X can take.

Taking the logarithm of L will make working with the likelihood easier and this does not change which parameters attain the maximum because the logarithm is a monotonic function.

$$\log L(\theta) = \log \prod_{x=1}^{k} \hat{P}_\theta(X = x)^{C(x)} = \sum_{x=1}^{k} C(x) \log \hat{P}_\theta(X = x) \tag{10}$$

This starts to look a bit like the definition of cross entropy. Note that if we divide $\log L(\theta)$ by the number of observations N, that this does not influence where $\log L(\theta)$ attains its maximum either:

$$\frac{1}{N} \log L(\theta) = \frac{1}{N} \sum_{x=1}^{k} C(x) \log \hat{P}_\theta(X = x) = \sum_{x=1}^{k} \frac{C(x)}{N} \log \hat{P}_\theta(X = x) \tag{11}$$

Let us call $\tilde{P}(X = x) = \frac{C(x)}{N}$ the empirical distribution. It is easy to see that $\tilde{P}(X = x)$ is indeed a distribution. That means that we can write eq 11 as the negative cross-entropy between the empirical distribution and our estimate $\hat{P}_\theta$:

$$\frac{1}{N} \log L(\theta) = \sum_{x=1}^{k} \tilde{P}(X = x) \log \hat{P}_\theta(X = x) \tag{12}$$

We can now substitute $\tilde{P}$ for P and $\hat{P}_\theta$ for Q in Gibbs' inequality. We multiply both sides with -1 to flip the direction of the inequality and arrive at:

$$\sum_{x=1}^{k} \tilde{P}(X = x) \log \tilde{P}(X = x) \geq \sum_{x=1}^{k} \tilde{P}(X = x) \log \hat{P}_\theta(X = x) = \frac{1}{N} \log L(\theta) \tag{13}$$

Note that the left side of the inequality is also a log likelihood of the data (divided by N), namely under the distribution $\tilde{P}$. This means that no matter what parameter values θ we choose for $\hat{P}_\theta$, it cannot give the data higher log likelihood than $\tilde{P}$. That is, the log likelihood (and therefore the likelihood) attains a maximum at $\tilde{P}$ and therefore $\tilde{P}(X = x) = \frac{C(x)}{N}$ is a maximum likelihood estimate. Actually this is also the only maximum (eq 8 holds with equality if and only if $\hat{P}_\theta = \tilde{P}$).